SMB2 Model-Based Testing

Nico Kicillof, Albert Lee

Microsoft
“BlueLine” Technical Document Testing Project

- 222 open protocol specifications tested
  - 22,847 pages
- ~10,000 document bugs
- 66,962 person days (250+ years)
- Tool dev center in Beijing
- Vendor test teams
  - Hyderabad: 250
  - Beijing: 100
Test-Driven Document Analysis

- Formalization
  - Validate consistency
  - Simulate implementation process

Test Suite
- Assertions
  - Checked against Windows (Accuracy)
Model-Based Testing

A Lightweight Formal Method
Requirements

- Gathered from technical document alone
  - Gatherers not protocol experts
  - Clean-room approach
  - Windows-specific behavior as separate requirements
  - Document testing
    - Not implementation

- Independent reviewers

- Main input for model design
The Value of Modeling

- Model-Based Testing vs. traditional testing comparison
  - Empirical SE Research group assessed statistical significance
  - Uniform sample from one testing site

- Modeling resulted in **42%** productivity gain
  - Overall, modeling saved 50 person-years (12,547 days)

- No formal methods background was required
  - Teams of junior engineers
  - Mostly vendors, fresh out of college
Test Representation

- Getting data on the wire and back
- Known problem for protocol testing (e.g. TTCN-3)
- Our Solution: Protocol Test Framework (PTF)
  - Test adapters written in a managed language (C#)
  - Custom support for dealing with protocols
  - Automatic data packet (de)serialization
    - Based on declarative data definitions
  - Protocol-specific logging capacity
    - Beacon packets
Test Adapters

Test Cases
(traditional or MBT)

Test Development

Adapter Interface

Adapter Development

Adapter Implementation

SUT
(System under test)

- Abstracts SUT functionality
- Contract between teams
  - Test case team
  - Adapter team

- Pluggable
  - Different server setups
  - Different transports
Spec Explorer
A Model-Based Testing Tool from Microsoft
Features

- Multiple modeling styles and languages
  - Programs, patterns, diagrams
- Asynchronous & non-deterministic systems
- State machine extraction from model program
- Test code generation from state machine
- Model composition
Spec Explorer Usage

- Internally used to test several Microsoft products
- External Early Adopter Program
- Shipping plans
  - MSDN DevLabs
  - Visual Studio 2010 Beta 2 timeframe
SMB2 Interoperability Test Suite
Test Suite Scope

- Validate the Technical Document against Windows for interoperability
  - Accuracy
  - Consistency
  - Usability

- Aimed at the Microsoft Communications Protocol Program
  - Only server behavior is tested
Test Approach (First Pass)

- Assumptions
  - Single connection
  - No compounded message
  - No nested shares
- Out of model scope
  - GSS-API authentication
  - Race conditions
  - Exhaustive flag combinations
- 121 scenarios
Test Suite Architecture

- Test generated by Model
- Traditional Test

- SMB2 Adapter Interface
- Traditional Adapter Interface
- SMB2 Adapter Core
- Traditional Adapter Core

- MS-SMB2 Infrastructure SDK

- Request Call
- Response Event

- Server Config Adapter Interface
- Server Config Adapter Core

- SMB2 Server
Demo

The SMB2 Test Suite
Conclusions

- Interoperability test suites developed for 200+ protocols
- Thousands of document bugs filed and fixed
- Test suite further usage
  - Regression testing
  - Third-party implementations (Plugfests)
- SMB2 test suite (first pass)
  - Document improvement
  - 80% requirement coverage
Questions

{nicok, albertle} @ microsoft . com
http://www.microsoft.com/protocols/